UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/526,393 | 03/02/2005 | Staffan Gestrelius | 0104-0508PUS1 | 3260 |

2292 7590 03/20/2009
BIRCH STEWART KOLASCH & BIRCH
PO BOX 747
FALLS CHURCH, VA 22040-0747

| EXAMINER | |
|---|---|
| HICKS, MICHAEL J | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2165 | |

| NOTIFICATION DATE | DELIVERY MODE |
|---|---|
| 03/20/2009 | ELECTRONIC |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

mailroom@bskb.com

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/526,393 | GESTRELIUS ET AL. | |
| | Examiner | Art Unit | |
| | Michael J. Hicks | 2165 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *23 December 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1,2,5 and 8-25* is/are pending in the application.

4a) Of the above claim(s) *17-20* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,5,8-16 and 21-25* is/are rejected.

7)☐ Claim(s) *5 and 13* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *3/2/2005*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

***Claim Rejections - 35 USC § 102***

1. Claims 1-2, 5, and 8-25 Pending.

Claims 17-20 Withdrawn.

Claims 3-4 and 6-7 Canceled.

***Response to Arguments***

2. Applicant's arguments, see remarks, filed 12/23/2008, with respect to the rejection(s) of claim(s) 1-3 under USC 101 and 102(b) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Shanmugasundaram et al. ("Relational Databases for Querying XML Documents: Limitations and Opportunities", Proc. Of the 25th VLDB Conference, Morgan Kaufmann, 1999 and referred to hereinafter as Shanmugasundaram).

At Applicants request, a signed copy of the IDS dated 3/2/2005 is included herewith.

***Claim Objections***

3. Claims 5 and 13 objected to because of the following informalities:

Claims 5 and 13 are currently indicated as depending from Claim 41. No such claim exists.

Appropriate correction is required.

***Claim Rejections - 35 USC § 112***

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 1-2, 5, 8-16, and 21-25 rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1, 22, 23, and 25 recites the limitation "metatype Object". There is insufficient antecedent basis for this limitation in the claim as metatypes are not previously mentioned in the Claims, and further, none of the n metadata types are defined in the independent claims. All other Claims are similarly rejected as depending from one of Claims 1, 22, 23, and 25.

Claim 1 recites the limitation "said branch table" in Line 16 and "the branch table" in Line 41. There is insufficient antecedent basis for this limitation in the claim as no branch table is disclosed in Claim 1. Claims 2, 5, 8-16, and 21 are similarly rejected as depending from Claim 1.

***Claim Rejections - 35 USC § 102***

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1-2, 5, 8-9, 12-14, 16, and 21-25 rejected under 35 U.S.C. 102(b) as being anticipated by Shanmugasundaram.

As per Claim 1, Shanmugasundaram discloses a computer program product having program code which program code is stored on a computer-readable storage medium and which, when executed on a processor is adapted to provide, a database of information objects and a database engine for managing said objects, characterized in that each information object comprises metadata for describing the information object and for defining a hierarchical structure of branches which constitutes said information object and which includes relations to other information objects (Figure 1 and Section 2.1 clearly disclose a information object which comprises metadata for describing the information object and for describing a hierarchical structure of branches (e.g. elements and sub-elements) which constitute said information object and which include relations to other information objects (e.g. the specification of a value for the <author> element allows for a relation with an <author> object, see id attributes in Section 2.2).), wherein an individual branch has: a metadata type, which is selected from a predetermined and limited set of n different metadata types and which represents one respective hierarchical level in said hierarchical structure (Figure 2 and Section 2.2. clearly indicate that the objects have a limited set metadata types, specified in the DTD, and which are identified as element or sub-elements (e.g., branches). Examiner further notes that each branch represents one hierarchical level in the hierarchical structure, as indicated in Figure 1, and the structured nature of the DTD in Figure 2.); a metadata value (Figure 1 and Section 2.1 clearly indicate that each branch has a metadata value which is either an attribute or sub-element.); and an arbitrary number of other

branches, of any hierarchical level strictly subordinate to the hierarchical level of said individual branch, connected to said individual branch as children thereof, said arbitrary number including zero branches (Figure 1 and Section 2.1 Clearly indicate that sub-elements must be strictly subordinate to the elements or sub-elements they are contained within. Examiner notes that the number of sub-elements is illustrated as arbitrary, in that an element may contain any number of sub elements, including zero.); said branch table having columns that correspond to the n different metadata types and each information object is stored in said database in the form of an n-field data structure is stored as a row in said branch table (The Second Column of the Second Page of Section 3.3 Clearly illustrates an example tuple for a book relation table, as defined by the relation schema found in Figure 10, corresponding to the example XML document from Figure 1. Examiner notes that each branch (e.g. element or sub-element) is represented as an attribute in the book relation table and the example tuple is a n-field data structure stored as a row in the table wherein n is the number of branches.); and a query function for searching said database (Figures 3 and 4 and section 2.3. clearly indicate a query function for searching the database.), wherein said query function is adapted to: accept a search query which is specified in a declarative language in accordance with said hierarchical structure of information objects in said database and which thus defines a branch hierarchy to search for (Figure 4 clearly indicates a search query specified in a declarative language (e.g. XML) which specifies a branch hierarchy (e.g. element hierarchy) to search for.); search said database so as to find information objects, if any, which contain a branch hierarchy that makes the one specified in said declarative language (Figure 4 clearly indicates a search query specified in a declarative language (e.g. XML) which specifies a branch hierarchy (e.g. element hierarchy) to search for. Examiner asserts that this also indicates that the database is searched.); and further adapted to comprise the steps of: a) receiving a search query (Figure 4 clearly indicates a search query specified in a declarative

language (e.g. XML) which specifies a branch hierarchy (e.g. element hierarchy) to search for.); b) transforming the search query into a search hierarchy of row searches and joins by: b1) generating a top join for joining all results at metatype Object (Examiner notes that this limitation is unclear as it is the subject of a 112 2nd rejection, and as such, will be interpreted to indicate that a join operation is placed at the root of a search strategy which joins the results of the searches performed on all root nodes/branches. Examiner asserts that this is taught on Page 1 of Section 4.1 wherein *"(a) first, the relation(s) corresponding to start of the root path expression(s) are identified and added to the from clause of the SQL query"*. Examiner notes that this excerpt indicates that all book (e.g. root node) objects will be searched and the results of these searches will be joined.); b2) traversing the branches of the search query top-down by b'1) if a particular branch has no children, creating a row search with the metadata values of all branches from the root down to the particular branch as conditions on metadata values in the row search (Examiner notes that this limitation is understood to indicate that the branch structure of the declarative language search query is traversed until a leaf branch is identified, at which point the values of the branches along the path to the leaf branch are used as search criteria. Examiner asserts that this is disclosed in Section 4.1 wherein the XML query is evaluated to identify that the sub-elements with no branches (e.g. booktitle, firstname, and lastname) are to be used as the search criteria (e.g. firstname and lastname are the desired results, and booktitle is the search requirement).); b'2) if a particular branch has more than one child, creating a join for joining row searches and joins at the metatype of the particular branch (Examiner notes that this is indicated in Section 4.1 which indicates *" (b) if necessary, the path expressions are translated to joins among relations (when elements are inlined, joins are not necessary). The SQL query generated in this fashion for the example query above is shown in Figure 18. Note that a join condition has been added to the where clause to link the book and author and a selection (A.parentCODE = 0, where 0 indicates that the parent of the author is a book) is performed on author to make sure that only authors reached through book are considered."*); b'3) adding a row

search created in step b'1) or a join created in step b'2), respectively, as a child of a preceding join (Examiner asserts that steps a) and b) of Section 4.1 outline a method for forming the search strategy and as such will continue adding row searches and joins as children of the previous operations to the search strategy in order to form the completed search strategy.); c) performing all the row searches from step b) so as to result in a set of rows in the branch table (Examiner strongly asserts that after the search query is converted, it will be executed therefor performing the operations outlined above including the row searches and joins.); d) performing the joins in a search hierarchy successfully starting from the bottom of the search hierarchy and ending with the top of the search hierarchy (Examiner strongly asserts that after the search query is converted, it will be executed therefor performing the operations outlined above including the row searches and joins.); and e) producing a result of the search query by retrieving all unique objects from the outcome of step d) (Examiner strongly asserts that running the search query will produce a result.).

As per Claim 2, Shanmugasundaram discloses each information object is stored in said database in the form of an n-field data structure for each particular branch that does not have any children, and wherein each field of the n-field data structure represents a respective branch that precedes said particular branch as its ancestor (The Second Column of the Second Page of Section 3.3 Clearly illustrates an example tuple for a book relation table, as defined by the relation schema found in Figure 10, corresponding to the example XML document from Figure 1. Examiner notes that each branch (e.g. element or sub-element) is represented as an attribute in the book relation table and the example tuple is a n-field data structure stored as a row in the table wherein n is the number of branches. Examiner further notes that each branch which has no

children is represented and each filed of the n filed data structure represents a predecessor element of these leaf elements.).

As per Claim 5, Shanmugasundaram discloses said declarative language is xml (The abstract of Shanmugasundaram clearly discloses the declarative language is XML.).

As per Claim 8, Shanmugasundaram discloses said metadata includes metadata attributes to respective branches of an information object (Figure 1 and Section 2.1 clearly disclose a information object which comprises metadata for describing the information object and for describing a hierarchical structure of branches (e.g. elements and sub-elements) which constitute said information object and which include relations to other information objects (e.g. the specification of a value for the <author> element allows for a relation with an <author> object, see id attributes in Section 2.2). Examiner notes that the metadata attributes to respective branches of the object.).

As per Claim 9, Shanmugasundaram discloses said query function is adapted to accept search queries with constraint conditions on said metadata attributes (Section 4.1 clearly discloses that a constraint is placed on the <booktitle> attribute.).

As per Claim 12, Shanmugasundaram discloses said metadata attributes include unit definitions for individual branches of an information object (Section 3.3, Page 2, column 2, states *"Each relation has an IF field that serves as the key of that relation. All relations corresponding to element nodes having a parent also have a parented field that serves as a foreign key..."* which clearly indicates that metadata attributes include unit definitions for individual branches of an information object).

As per Claim 13, Shanmugasundaram discloses a branch of a predetermined first metadata type in an individual information object defines a relation from said individual information object to another information object (Section 2.2 clearly indicates that objects have id attributes which uniquely identify them and which are used to reference one object from another object. Examiner notes that as this id metadata is an attribute, it can be used to match objects in a search query of the query function.) and wherein said query function is adapted to search said database to find said another information object by matching the metadata thereof with the metadata included in said branch of a predetermined first metadata type value in said individual information object (Section 2.2 clearly indicates that objects have id attributes which uniquely identify them and which are used to reference one object from another object. Examiner notes that as this id metadata is an attribute, it can be used to match objects in a search query of the query function.).

As per Claim 14, Shanmugasundaram discloses a branch of a predetermined second metadata type, other than said first metadata type, in an individual information object allows another information object to define a reverse relation to said individual information object, in the form of a branch of said predetermined first metadata type in said another information object (Section 2.2 clearly indicates that objects have id attributes which uniquely identify them and which are used to reference one object from another object. Examiner notes that as this id metadata is an attribute, it can be used to match objects in a search query of the query function. Examiner notes that this may be a reverse relation in that it uniquely identifies the object.), and wherein said query function is adapted to find said another information object by matching the metadata included in said branch of a predetermined second metadata

type in said individual information object with the metadata included in said branch of a predetermined first metadata type in said another information object (Section 2.2 clearly indicates that objects have id attributes which uniquely identify them and which are used to reference one object from another object. Examiner notes that as this id metadata is an attribute, it can be used to match objects in a search query of the query function. Examiner notes that this may be a reverse relation in that it uniquely identifies the object.).

As per Claim 16, Shanmugasundaram discloses specifying, through a man-machine interface of a computer, a search query in a declarative language in accordance with said hierarchical structure of information objects in said database (Section 2.3 clearly indicates that search queries in a declarative language in accordance with said hierarchical structure of information objects may be specified.); submitting said search query to said database engine through said computer (Section 4.1 clearly indicates that these search queries may be submitted to the database); receiving a result of said search query at said computer (Sections 4-4.1 clearly indicates that search results are retrieved from the database to be presented to the user); and presenting said result through said man-machine interface (Sections 4-4.1 clearly indicates that search results are retrieved from the database to be presented to the user).

As per Claim 21, Shanmugasundaram discloses for an assumed individual information object, of for all branches in the assumed information object that are of a first predetermined metadata type, said first predetermined metadata type allowing a relation to be defined from the assumed information object to another information object

(Section 2.2 clearly indicates that objects have id attributes which uniquely identify them and which are used to reference one object from another object. Examiner notes that as this id metadata is an attribute, it can be used to match objects in a search query of the query function.): forming a query based on the metadata of the branch, searching the database with the query, and collecting, as a result of the searching step, all information objects, if any, that the assumed information object has a relation to (Section 2.2 clearly indicates that objects have id attributes which uniquely identify them and which are used to reference one object from another object. Examiner notes that as this id metadata is an attribute, it can be used to match objects in a search query of the query function.); and for all branches in the assumed information object that are of a second predetermined metadata type, other than said first predetermined metadata type, said second predetermined metadata type allowing a reverse relation to be defined from another information object to the assumed information object (Section 2.2 clearly indicates that objects have id attributes which uniquely identify them and which are used to reference one object from another object. Examiner notes that as this id metadata is an attribute, it can be used to match objects in a search query of the query function. Examiner notes that this may be a reverse relation in that it uniquely identifies the object.): forming a query based on the metadata of the branch, searching the database with the query, and collecting, as a result of the searching step, all information objects, if any, that have a relation to the assumed information object (Section 2.2 clearly indicates that objects have id attributes which uniquely identify them and which are used to reference one object from another object. Examiner notes that as this id metadata is an attribute, it can be used to match objects in a search query of the query function. Examiner notes that this may be a reverse relation in that it uniquely identifies the object.).

As per Claim 22, Shanmugasundaram discloses a computer program product having program code, which program code is stored on a computer-readable storage medium and which, when executed on a processor is adapted to provide a database of information objects and a database engine for managing said information objects, characterized in that each information object comprises metadata for describing the information object and for defining a hierarchical structure of branches which constitutes said information object and which includes relations to other information objects (Figure 1 and Section 2.1 clearly disclose a information object which comprises metadata for describing the information object and for describing a hierarchical structure of branches (e.g. elements and sub-elements) which constitute said information object and which include relations to other information objects (e.g. the specification of a value for the <author> element allows for a relation with an <author> object, see id attributes in Section 2.2).), wherein an individual branch has: a metadata type, which is selected from a predetermined and limited set of n different metadata types and which represents one respective hierarchical level in said hierarchical structure (Figure 2 and Section 2.2. clearly indicate that the objects have a limited set metadata types, specified in the DTD, and which are identified as element or sub-elements (e.g.. branches). Examiner further notes that each branch represents one hierarchical level in the hierarchical structure, as indicated in Figure 1, and the structured nature of the DTD in Figure 2.); a metadata value (Figure 1 and Section 2.1 clearly indicate that each branch has a metadata value which is either an attribute or sub-element.); and an arbitrary number of other branches, of any hierarchical level strictly subordinate to the hierarchical level of said individual branch, connected to said individual branch as children thereof, said arbitrary number including zero branches (Figure 1 and Section 2.1 Clearly indicate that sub-elements must be strictly subordinate to the elements or sub-elements they are contained within. Examiner notes that the number of sub-elements is illustrated as arbitrary, in that an

element may contain any number of sub elements, including zero.), wherein the computer program product is further adapted to: a) receiving a search query (Figure 4 clearly indicates a search query specified in a declarative language (e.g. XML) which specifies a branch hierarchy (e.g. element hierarchy) to search for.); b) transforming the search query into a search hierarchy of row searches and joins by b1) generating a top join for joining all results at metatype Object (Examiner notes that this limitation is unclear as it is the subject of a 112 2nd rejection, and as such, will be interpreted to indicate that a join operation is placed at the root of a search strategy which joins the results of the searches performed on all root nodes/branches. Examiner asserts that this is taught on Page 1 of Section 4.1 wherein *"(a) first, the relation(s) corresponding to start of the root path expression(s) are identified and added to the from clause of the SQL query"*. Examiner notes that this excerpt indicates that all book objects will be searched and the results of these searches will be joined.); b2) traversing the branches of the search query top down by b'1) if a particular branch has no children, creating a row search with the metadata values of all branches from the root down to the particular branch as conditions on metadata values in the row search (Examiner notes that this limitation is understood to indicate that the branch structure of the declarative language search query is traversed until a leaf branch is identified, at which point the values of the branches along the path to the leaf branch are used as search criteria. Examiner asserts that this is disclosed in Section 4.1 wherein the XML query is evaluated to identify that the sub-elements with no branches (e.g. booktitle, firstname, and lastname) are to be used as the search criteria (e.g. firstname and lastname are the desired results, and booktitle is the search requirement).); b'2) if a particular branch has more than one child, creating a join for joining row searches and joins at the metatype of the particular branch (Examiner notes that this is indicated in Section 4.1 which indicates *" (b) if necessary, the path expressions are translated to joins among relations (when elements are inlined, joins are not necessary). The SQL query generated in this fashion for the example query above is shown in Figure 18. Note that a join condition has been added to the where clause to link the*

*book and author and a selection (A.parentCODE = 0, where 0 indicates that the parent of the author is a book) is performed on author to make sure that only authors reached through book are considered.*"); b'3) adding a row search created in step b'1) or a join created in step b'2), respectively, as a child of a preceding join (); c) performing all the row searches from step b) so as to result in a set of rows in a branch table (Examiner strongly asserts that after the search query is converted, it will be executed therefor performing the operations outlined above including the row searches and joins.); d) performing the joins in the search hierarchy successively starting from the bottom of the search hierarchy and ending with the top of the search hierarchy (Examiner strongly asserts that after the search query is converted, it will be executed therefor performing the operations outlined above including the row searches and joins.); and e) producing a result of the search query by retrieving all unique information objects from the outcome of step d) (Examiner strongly asserts that running the search query will produce a result.).

As per Claim 23, Shanmugasundaram discloses a computer program product having program code, which program code is stored on a computer-readable storage medium and which, when executed on a processor is adapted to provide a database of information objects and a database engine for managing said information objects, characterized in that - each information object comprises metadata for describing the information object and for defining a hierarchical structure of branches which constitutes said information object and which includes relations to other information objects (Figure 1 and Section 2.1 clearly disclose a information object which comprises metadata for describing the information object and for describing a hierarchical structure of branches (e.g. elements and subelements) which constitute said information object and which include relations to other information

objects (e.g. the specification of a value for the <author> element allows for a relation with an <author> object, see id attributes in Section 2.2).), wherein an individual branch has: - a metadata type, which is selected from a predetermined and limited set of n different metadata types and which represents one respective hierarchical level in said hierarchical structure (Figure 2 and Section 2.2. clearly indicate that the objects have a limited set metadata types, specified in the DTD, and which are identified as element or sub-elements (e.g.. branches). Examiner further notes that each branch represents one hierarchical level in the hierarchical structure, as indicated in Figure 1, and the structured nature of the DTD in Figure 2.); - a metadata value (Figure 1 and Section 2.1 clearly indicate that each branch has a metadata value which is either an attribute or sub-element.); and - an arbitrary number of other branches, of any hierarchical level strictly subordinate to the hierarchical level of said individual branch, connected to said individual branch as children thereof, said arbitrary number including zero branches (Figure 1 and Section 2.1 Clearly indicate that sub-elements must be strictly subordinate to the elements or sub-elements they are contained within. Examiner notes that the number of sub-elements is illustrated as arbitrary, in that an element may contain any number of sub elements, including zero.), wherein the computer program product is further adapted to: a) receiving a search query (Figure 4 clearly indicates a search query specified in a declarative language (e.g. XML) which specifies a branch hierarchy (e.g. element hierarchy) to search for.); b) transforming the search query into a search hierarchy of row searches and joins by b1) generating a top join for joining all results at metatype Object (Examiner notes that this limitation is unclear as it is the subject of a 112 2nd rejection, and as such, will be interpreted to indicate that a join operation is placed at the root of a search strategy which joins the results of the searches performed on all root nodes/branches. Examiner asserts that this is taught on Page 1 of Section 4.1 wherein *"(a) first, the relation(s) corresponding to start of the root path expression(s) are identified and added to the from clause of the SQL query"*. Examiner notes that this

excerpt indicates that all book objects will be searched and the results of these searches will be joined.); b2) traversing the branches of the search query topdown by performing row searches (Examiner notes that this limitation is understood to indicate that the branch structure of the declarative language search query is traversed until a leaf branch is identified, at which point the values of the branches along the path to the leaf branch are used as search criteria. Examiner asserts that this is disclosed in Section 4.1 wherein the XML query is evaluated to identify that the sub-elements with no branches (e.g. booktitle, firstname, and lastname) are to be used as the search criteria (e.g. firstname and lastname are the desired results, and booktitle is the search requirement).); c) performing all the row searches from step b) so as to result in a set of rows in a branch table (Examiner strongly asserts that after the search query is converted, it will be executed therefor performing the operations outlined above including the row searches and joins.); d) performing the joins in the search hierarchy successively starting from the bottom of the search hierarchy and ending with the top of the search hierarchy (Examiner strongly asserts that after the search query is converted, it will be executed therefor performing the operations outlined above including the row searches and joins.); and e) producing a result of the search query by retrieving all unique information objects from the outcome of step d) (Examiner strongly asserts that running the search query will produce a result.).

As per Claim 24, Shanmugasundaram discloses said row searches comprises at least, if a particular branch has no children, creating a row search with the metadata values of all branches from the root down to the particular branch as conditions on metadata values in the row search (Examiner notes that this limitation is understood to indicate that the branch structure of the declarative language search query is traversed until a leaf branch is identified, at which point the values of the branches along the path to the leaf branch are used as search

criteria. Examiner asserts that this is disclosed in Section 4.1 wherein the XML query is evaluated to identify that the sub-elements with no branches (e.g. booktitle, firstname, and lastname) are to be used as the search criteria (e.g. firstname and lastname are the desired results, and booktitle is the search requirement).).

As per Claim 25, Shanmugasundaram discloses a computer program product having program code, which program code is stored on a computer-readable storage medium and which, when executed on a processor is adapted to provide a database of information objects and a database engine for managing said information objects, characterized in that each information object comprises metadata for describing the information object and for defining a hierarchical structure of branches which constitutes said information object and which includes relations to other information objects (Figure 1 and Section 2.1 clearly disclose a information object which comprises metadata for describing the information object and for describing a hierarchical structure of branches (e.g. elements and sub-elements) which constitute said information object and which include relations to other information objects (e.g. the specification of a value for the <author> element allows for a relation with an <author> object, see id attributes in Section 2.2).), wherein an individual branch has: a metadata type, which is selected from a predetermined and limited set of n different metadata types and which represents one respective hierarchical level in said hierarchical structure (Figure 2 and Section 2.2. clearly indicate that the objects have a limited set metadata types, specified in the DTD, and which are identified as element or sub-elements (e.g.. branches). Examiner further notes that each branch represents one hierarchical level in the hierarchical structure, as indicated in Figure 1, and the structured nature of the DTD in Figure 2.); a metadata value (Figure 1 and Section 2.1 clearly indicate that each branch has a metadata value which is either an attribute or sub-element.); and an arbitrary

number of other branches, of any hierarchical level strictly subordinate to the hierarchical level of said individual branch, connected to said individual branch as children thereof, said arbitrary number including zero branches (Figure 1 and Section 2.1 Clearly indicate that sub-elements must be strictly subordinate to the elements or sub-elements they are contained within. Examiner notes that the number of sub-elements is illustrated as arbitrary, in that an element may contain any number of sub elements, including zero.), wherein the computer program product is further adapted to: a) receiving a search query (Figure 4 clearly indicates a search query specified in a declarative language (e.g. XML) which specifies a branch hierarchy (e.g. element hierarchy) to search for.); b) transforming the search query into a search hierarchy of searches and joins by b1) generating a top join for joining all results at metatype Object (Examiner notes that this limitation is unclear as it is the subject of a 112 2nd rejection, and as such, will be interpreted to indicate that a join operation is placed at the root of a search strategy which joins the results of the searches performed on all root nodes/branches. Examiner asserts that this is taught on Page 1 of Section 4.1 wherein *"(a) first, the relation(s) corresponding to start of the root path expression(s) are identified and added to the from clause of the SQL query"*. Examiner notes that this excerpt indicates that all book objects will be searched and the results of these searches will be joined.); b2) traversing the branches of the search query topdown by performing searches (Examiner notes that this limitation is understood to indicate that the branch structure of the declarative language search query is traversed until a leaf branch is identified, at which point the values of the branches along the path to the leaf branch are used as search criteria. Examiner asserts that this is disclosed in Section 4.1 wherein the XML query is evaluated to identify that the sub-elements with no branches (e.g. booktitle, firstname, and lastname) are to be used as the search criteria (e.g. firstname and lastname are the desired results, and booktitle is the search requirement).), wherein said searches comprises, if a particular branch has no children, creating a search with the

metadata values of all branches from the root down to the particular branch as conditions on metadata values of information objects in branches of the search (Examiner notes that this limitation is understood to indicate that the branch structure of the declarative language search query is traversed until a leaf branch is identified, at which point the values of the branches along the path to the leaf branch are used as search criteria. Examiner asserts that this is disclosed in Section 4.1 wherein the XML query is evaluated to identify that the sub-elements with no branches (e.g. booktitle, firstname, and lastname) are to be used as the search criteria (e.g. firstname and lastname are the desired results, and booktitle is the search requirement).); c) performing all the searches from step b) (Examiner strongly asserts that after the search query is converted, it will be executed therefor performing the operations outlined above including the row searches and joins.); d) performing the joins in the search hierarchy successively starting from the bottom of the search hierarchy and ending with the top of the search hierarchy (Examiner strongly asserts that after the search query is converted, it will be executed therefor performing the operations outlined above including the row searches and joins.); and e) producing a result of the search query by retrieving all unique information objects from the outcome of step d) (Examiner strongly asserts that running the search query will produce a result.).

### *Claim Rejections - 35 USC § 103*

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claim 10 rejected under 35 U.S.C. 103(a) as being unpatentable over Shanmugasundaram in view of Bohlen et al. ("Point-versus interval-based temporal data models", Proceedings of the 14th International Conference on Data Engineering, Pages 192-200, 1998, IEEE and referred to hereinafter as Bohlen).

As per Claim 10, Shanmugasundaram fails to disclose said metadata attributes include timestamps on individual branches of an object.

Bohlen discloses said metadata attributes include timestamps on individual branches of an object. (See Bohlen, Abstract, Section 2.1 and Figure 1).

It would have been obvious to one skilled in the art at the time of Applicants intention to modify the art of Shanmugasundaram with the art of Bohlen to disclose discloses said metadata attributes include timestamps on individual branches of an object with the motivation that the association of timestamps with various data items such as tuples or attribute values is fundamental to the management of time-varying information (Bohlen, Abstract).

10. Claim 11 rejected under 35 U.S.C. 103(a) as being unpatentable over Shanmugasundaram in view of Damiani et al. ("ACM Transactions on Information and System Security", Vol. 5, No. 2, Pages 169-202, May 2002, ACM and referred to hereinafter as Damiani).

As per Claim 11, Shanmugasundaram fails to disclose said metadata attributes include access rights to individual branches of an object.

Damiani discloses said metadata attributes include access rights to individual branches of an object (Page 187, Section 6.1 clearly indicates that *"Each access authorization states whether a subject can (or cannot) access an element/attribute (or set of them)"* which state that access controls may be as fine grained as attribute or element/branch level.).

It would have been obvious to one skilled in the art at the time of Applicants intention to modify the art of Shanmugasundaram with the art of Damiani to disclose said metadata attributes include access rights to individual branches of an object with the motivation to present an access control model to protect information distributed on the Web that, by exploiting XML's own capabilities, allows the definition and enforcement of access restrictions directly on the structure and content of the documents (Damini, Abstract).

11. Claim15 rejected under 35 U.S.C. 103(a) as being unpatentable over Shanmugasundaram.

As per Claim 15, Shanmugasundaram fails to specifically disclose n = 6 and said set of metadata types consists of {Object, Relation, Key, Attribute, Type, and Value}, metadata type Object representing the root level of said hierarchical structure.

However, it is obvious from the disclosure of Shanmugasundaram that the object may have exactly six metadata types inasmuch as an object can exist with only six meaningful pieces of information to catalogue. Furthermore the names of these metadata types are considered to be nonfunctional descriptive material as the names of

the metadata types do not affect their function. Examiner further notes that above in regard to Claim 1, the assumption was made that the metatype Object was a reference to the root due to the 112 2nd rejection, and as such it is considered formally here, and for the same reasons as stated above, that the metadata type Object signifies the root node indicated by Shanmugasundaram.

It would have been obvious to one skilled in the art at the time of Applicants intention to modify the art of Shanmugasundaram as such to disclose n = 6 and said set of metadata types consists of {Object, Relation, Key, Attribute, Type, and Value}, metadata type Object representing the root level of said hierarchical structure with the motivation of allowing the number of metadata types for an object reflect the needs of the object.

12. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

***Points of Contact***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Hicks whose telephone number is (571) 272-2670. The examiner can normally be reached on Monday - Friday 9:00a - 5:30p.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Christian Chace can be reached on (571) 272-4190. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/John R. Cottingham/
Supervisory Patent Examiner, Art Unit 2167

Michael J Hicks
Art Unit 2165
Phone: (571) 272-2670
Fax: (571) 273-2670